

Parallel Multiscale Algorithms for Astrophysical Fluid Dynamics Simulations

Final Report for NASA Grant NAG 5-2493

Principal Investigator

Michael L. Norman
Department of Astronomy & NCSA
University of Illinois at Urbana-Champaign
Urbana, IL 61801
norman@ncsa.uiuc.edu

<http://zeus.ncsa.uiuc.edu:8080/AMR/amr.html>

November 20, 1997

1 Objective

Our goal is to develop software libraries and applications for astrophysical fluid dynamics simulations in multidimensions that will enable us to resolve the large spatial and temporal variations that inevitably arise due to gravity, fronts and microphysical phenomena. The software must run efficiently on parallel computers and be general enough to allow the incorporation of a wide variety of physics. Cosmological structure formation with realistic gas physics is the primary application driver in this work. Accurate simulations of e.g. galaxy formation require a spatial dynamic range (i.e., ratio of system scale to smallest resolved feature) of 10^4 or more in three dimensions in arbitrary topologies. We take this as our technical requirement. We have achieved, and in fact, surpassed these goals.

Grid Hierarchy

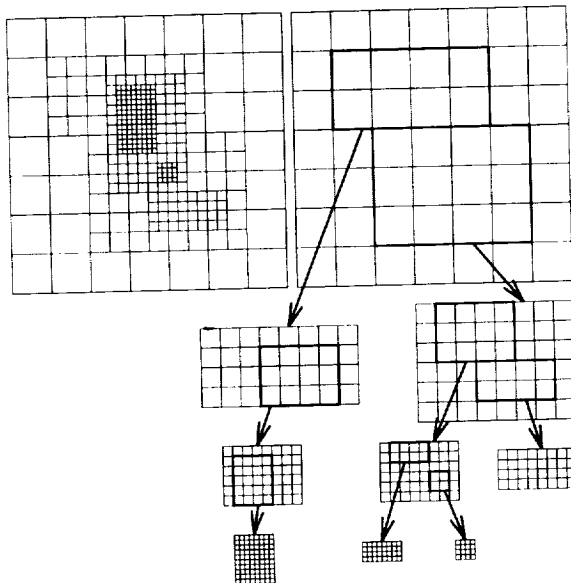


Figure 1: Illustration of an AMR grid hierarchy.

2 Approach

As we are interested in Eulerian grid-based schemes for solving the fluid equations, we adopt the structured adaptive mesh refinement (AMR) algorithm of Berger and Colella (1989). AMR uses a logical hierarchy of grids of various levels of refinement to achieve high resolution locally (cf. Fig. 1.) Generally, the computation begins with a single coarsely resolved grid. Then, as fine scale structure develops, subgrids are automatically introduced or deleted as the solution evolves. This is done adaptively and automatically.

We have adhered closely to their strategies and algorithms regarding subgridding, clustering and flux conservation. In particular, we avoid rotating and overlapping subgrids, which simplifies intergrid interactions. We have developed two applications, described below, which place no constraints on the refinement factor, the number or shape of the subgrids, or the number of levels of refinement.

Our target parallel architectures are RISC-based symmetric multiprocessors (SMP)

and distributed shared memory (DSM) machines. Because the AMR algorithm is inherently sequential across levels of refinement and parallel within a level, we adopt a coarse grain approach wherein each subgrid within a given level of refinement is a parallel thread of execution. Preliminary studies, presented below, indicate good parallel efficiency up to eight processors on Silicon Graphics MIPS R10000-based machines.

3 Results

3.1 Cosmological AMR

We have developed an application of Berger & Colella's (1989) structured adaptive mesh refinement algorithm to hydrodynamic cosmology. The essential difficulty here is to properly couple algorithms for modeling the collisionless components (dark matter and stars) and self gravity to the hydrodynamics. Our solution to this problem is briefly described in Bryan & Norman (1997a); an application to the simulation of an X-ray cluster of galaxies can be found in Bryan & Norman (1997b).

The gaseous component is evolved with the Piecewise Parabolic Method (Colella & Woodward 1984) suitably modified to account for cosmic expansion, gravitational accelerations, and the extreme Mach numbers encountered in cosmological structure formation (Bryan et al. 1995). With regard to the latter, we have developed a dual energy formulation that ensures accurate temperatures in low density regions while guaranteeing energy conservation across shock fronts. The collisionless component is evolved with the standard particle-mesh (PM) algorithm which we have generalized to an adaptive grid hierarchy. A particular feature of our algorithm is that we maintain only one list of particles; i.e., a hierarchy of particle masses is not employed, as in the hierarchical particle mesh (HPM) algorithm developed by Villumsen (1989). Rather, particles are assigned to the finest subgrid which contain them. In this regard, our algorithm can be construed as a particle hierarchical mesh (PHM) algorithm. Gravity is solved on each subgrid using FFTs with isolated boundary conditions interpolated from the parent grid. The gravitational potential on the root grid is generally computed assuming periodic boundary conditions.

The same grid hierarchy is used for the hydrodynamics, the PM density assignment and force interpolation, and the gravitational field solve. Subgrids are introduced based on a local overdensity criterion rather than the cononical truncation error criterion on Berger & Colella (1989). Furthermore, although our implementation allows an arbitrary integer refinement factor, we find that a refinement factor of two provides the best results for cosmological simulations where a comensurate mass resolution in both the gas and dark matter must be maintained for accuracy sake.

The AMR framework is implemented in C++ to handle the recursive logic and mem-

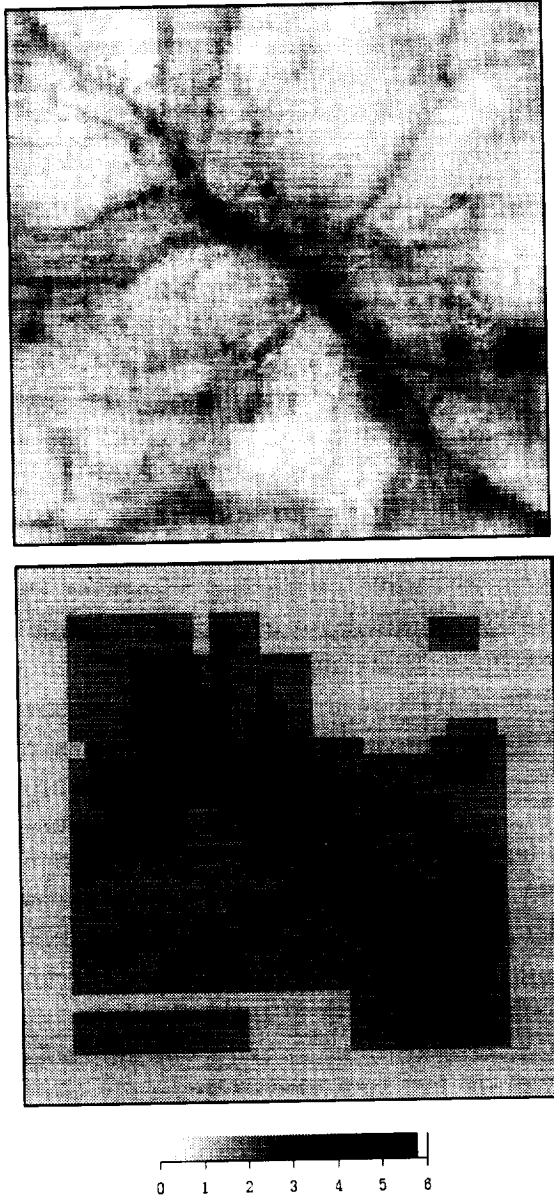


Figure 2: Application of adaptive mesh refinement to the formation of a cluster of galaxies, seen here at redshift $= 2$. The logarithm of the dark matter surface density (top) and the projected grids (bottom), shaded according to level of refinement. In order to increase the contrast of the three level 6 grids, they have been colored white. The spatial scale is 32 Mpc on a side. Cell resolution at level l is $1\text{Mpc}/2^{l+1}$.

ory management functions, while the computationally intensive tasks are programmed in F77.

Fig. 2 shows an application of this code to the simulation of the formation of an X-ray cluster of galaxies in a standard cold dark matter cosmogony. The box size is 64 comoving Mpc. The root grid is 64^3 cells on a side. As structure forms, refined subgrids are introduced automatically wherever needed as defined by our overdensity criterion. The figure shows the projected dark matter density at a redshift of 2, as well as a “shadowgraph” of the grid hierarchy at that time. Altogether, seven levels of refinement and over 400 subgrids are used at this epoch to resolve the structure formation process. A maximum effective resolution of $8,192^3$ is achieved in the finest subgrids, which coincide with the centers of high density knots. In a separate simulation of galaxy formation, we have achieved a maximum effective resolution of $16,384^3$ in the protogalaxies, exceeding the technical requirement we set out for ourselves.

3.2 HAMR

Concurrently with the project described above, we have also developed the HAMR (Hierarchical Adaptive Mesh Refinement) System, which is a general purpose, flexible, extensible, portable software system for simplifying the construction of structured AMR applications based on the Berger & Colella (1989) algorithm. HAMR is a product of the computer science Ph. D. thesis of Henry Neeman (Neeman 1996; Neeman & Norman 1997).

HAMR’s autonomy and generality arise from its *slot-and-fill* design which provides function pointers to basic AMR methods in the HAMR library such as interpolators, error estimators, clustering, etc., or to user-supplied routines, such as initialization and physics solvers. Not only are algorithms available on this basis, but so is data: the application scientist can declare an arbitrary number of fields, lists and other such simple data structures. These declarations can apply to the grid hierarchy as a whole, to a specific level of resolution, or to an individual grid. In addition to providing slots for declaring data and methods (subroutines), HAMR also provides one more crucial set of slots: attributes. Attributes describe the spatio-temporal extent of data items and their interrelationships with other data and methods. HAMR is autonomous in the sense that, given a set of these declarations, the system can create, initialize and run an application whose fundamental characteristics—that is, its variables and method, and their interrelationships—are known at compile time, without requiring the application researcher to develop the code that accesses these items. In essence, HAMR encapsulates the application subroutines, insulating the application scientist from the complicated and cumbersome details of data management and AMR implementation.

Fig. 3 shows the principal software components of the HAMR system. HAMR supports multiple grid geometries, variable centerings, and dimensionality (up to six.)

HAMR Software Components

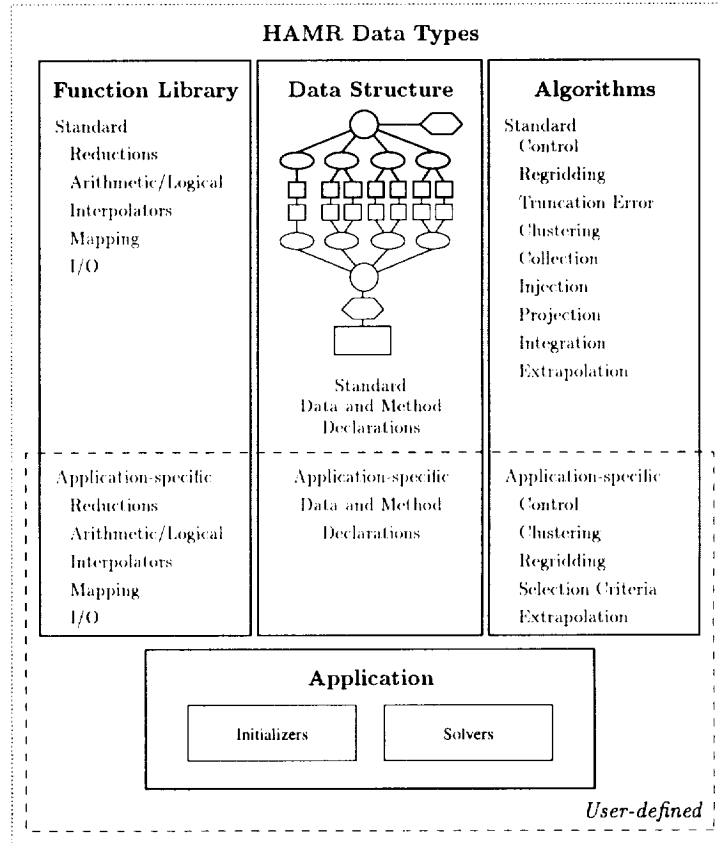


Figure 3: Block diagram of the HAMR system.

HAMR is implemented in ANSI C, and is thus portable. Using HAMR, we have built applications for the 1D wave equation and 3D gas dynamics in only a few days. More information and some sample results can be found at the HAMR web site:

<http://zeus.ncsa.uiuc.edu:8080/~hneeman/hamr.html>

3.3 Parallelization

Both the HAMR system and the cosmological AMR code implement a shared memory model. This is because their target architectures are shared memory machines, which include hardware distributed shared memory (DSM) machines such as the HP/Convex Exemplar and the SGI/Cray Origin2000. With the use of a software DSM library like Treadmarks (Amza et al. 1996.), we can in principle run our codes on distributed memory MPPs and clusters of workstations.

Like multigrid, AMR provides no opportunity for parallelism on the hierarchy as a whole: higher order time accuracy requires sequentially updating each level in the hierarchy in a “W-cycle.” However, we can exploit coarse-grained parallelism within a given level, where there may be dozens of grids. Alternatively, we can exploit fine-grained (i.e., loop-level) parallelism within a given grid. We have pursued the former approach, as we find this minimizes overhead and yields better results.

Coarse-grained parallelization is accomplished within the cosmology code by looping over the grids within a level and instructing the compiler to execute these iterations concurrently. Data dependencies are eliminated by setting boundary values on each grid in a previous sequential step. Fig. 4 shows the parallel speedup for the cluster simulation in Fig. 2; Figs. 5 and 6 shows the distribution of grids and work across levels of the hierarchy. We typically find good speedup to 8 processors, levelling off. This is due to the fact that there is relatively little work to do at both ends of the hierarchy (i.e., at the root grid and at the “leaf” grids.) At present, we use a single, rather small root grid to initialize the calculation. Parallel performance would improve considerably if we achieved the same maximum resolution using fewer levels of refinement but a larger root grid (e.g., 256^3), and distributed it across processors. This efficiency would come at the expense of a higher memory and CPU requirement.

We are parallelizing HAMR in the same coarse-grained manner by linking to the HPC++ parallel thread library developed by Dennis Gannon at the Dept. of Computer Science, Indiana University. This library is being implemented on many high performance computer systems, including DSM, SMP, MPP and workstation clusters. We have no results to report as yet.

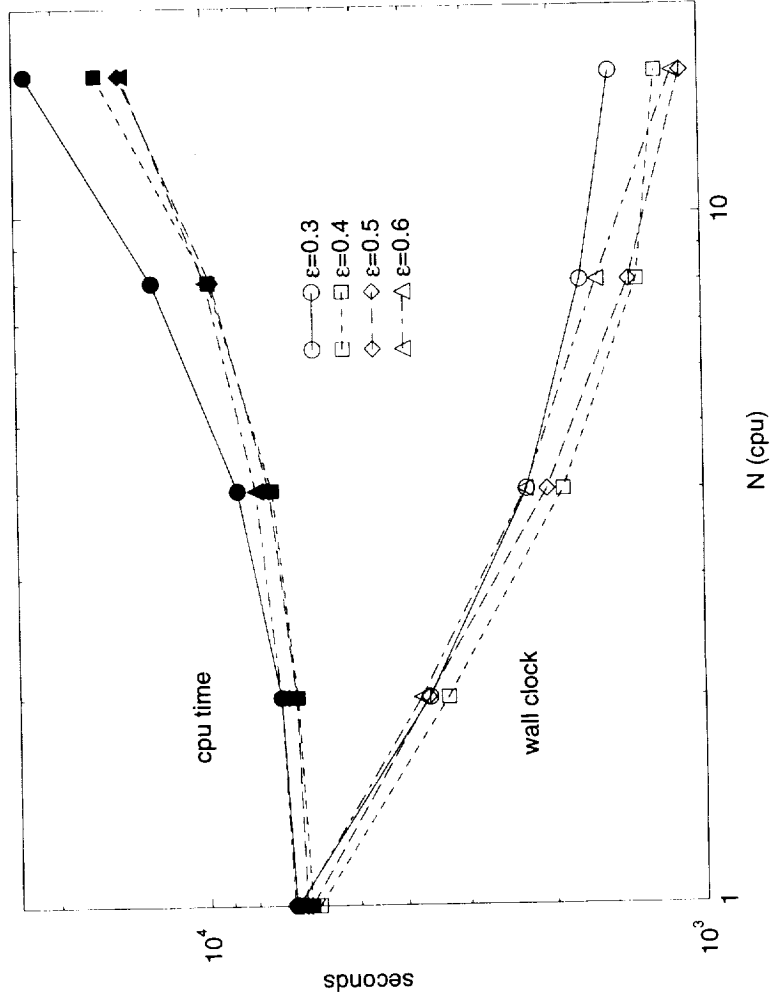


Figure 4: Parallel speedup of the cosmological AMR code on an Origin2000 computer.

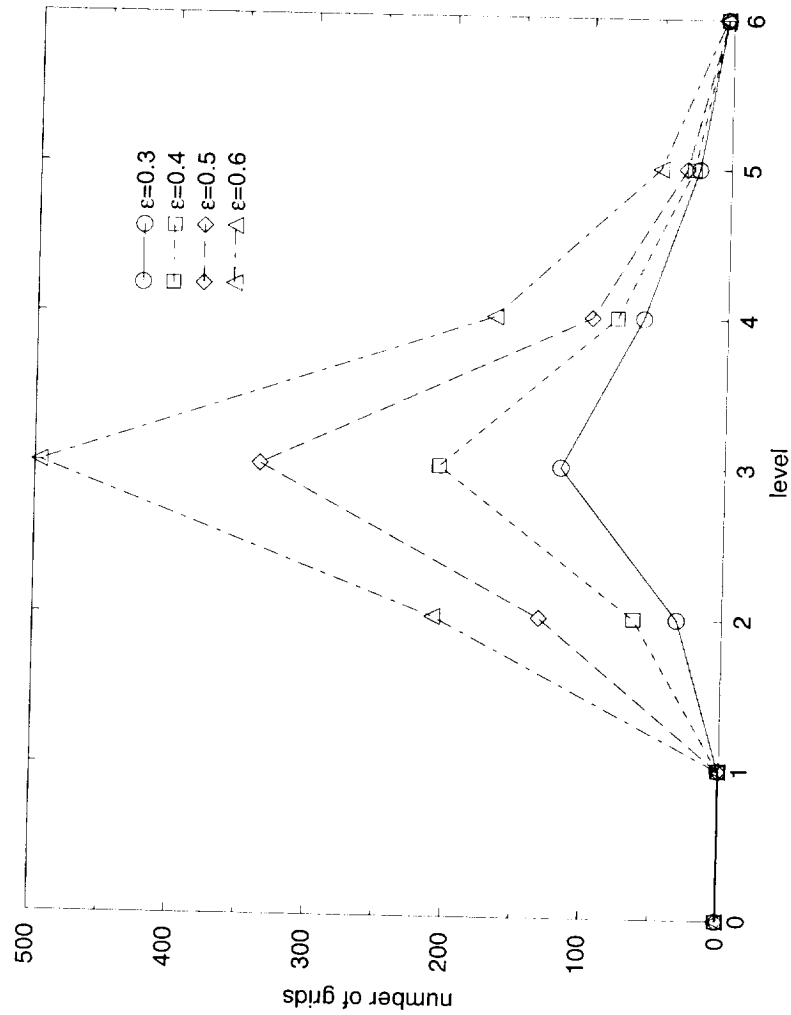


Figure 5: Distribution of the number of grids in the AMR hierarchy versus level for the example shown in Figure 2 for various values for the accuracy parameter ϵ .

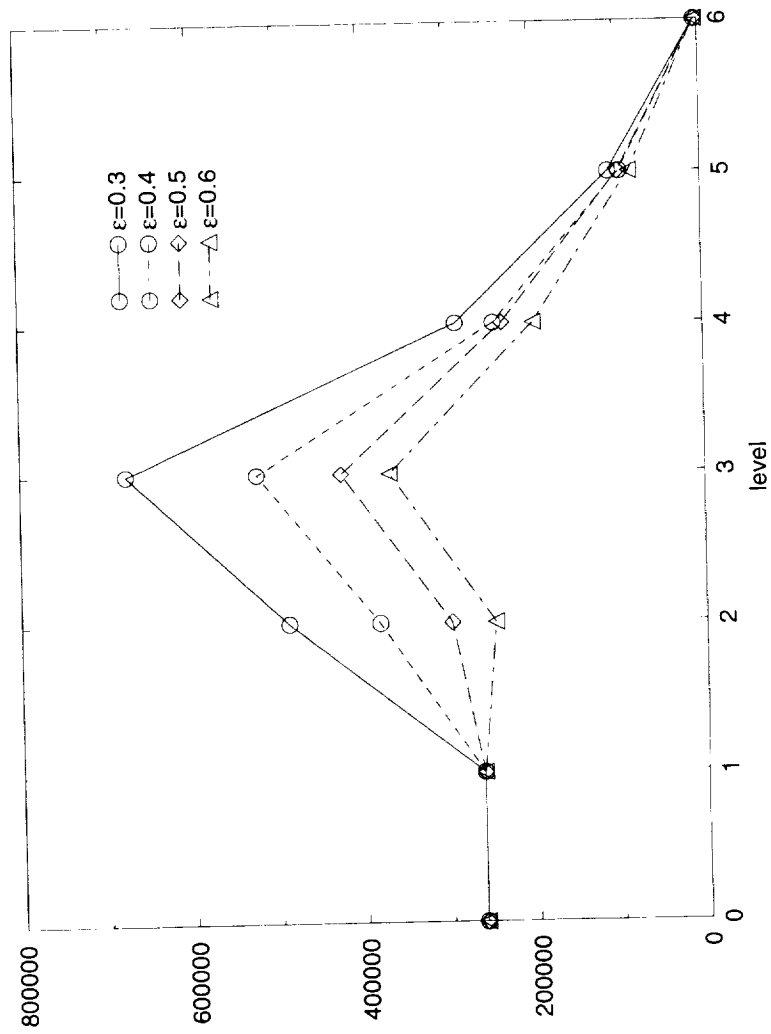


Figure 6: Distribution of the number of active cells in the AMR hierarchy versus level for the example shown in Figure 2 for various values for the accuracy parameter ϵ .

4 Future Work

We will continue developing, optimizing and applying our two codes to problems in astrophysics and cosmology. The cosmology AMR code underpins three currently funded NASA Astrophysics Theory Program grants. In one project (PI Anatoly Klypin), we will model the formation and evolution of X-ray clusters of galaxies including the effects of radiative cooling, galaxy formation and feedback. We are applying the cosmological AMR code to simulate at high spatial resolution a complete sample of ~ 100 X-ray clusters in several viable CDM-like models of structure formation. Despite the large box size (256 Mpc/h), we will be able to achieve 15 kpc/h resolution within each cluster. This will allow an accurate determination of the cluster halo luminosity function and temperature function, as well as detailed cluster maps showing substructure and cooling flows. In a second project (PI Michael Norman), we will use the AMR code to simulate the formation of the first structures in CDM-like models, and their contribution to the reionization and metal enrichment of the IGM. In a third (PI Piero Madau), we will simulate the epoch of reionization including expansion of localized H II regions, their percolation, and the evolution of the UV background. Greg Bryan is using the AMR code to simulate galaxy formation for comparison with recent HST observations of high redshift galaxies.

On the computational side, we will scale our simulations to larger problem sizes and numbers of processors, exploring different data distribution and load balancing strategies.

5 Publications Resulting from Grant

- Bryan, G. L. & Norman, M. L. 1997a. “A Hybrid Application for Cosmology and Astrophysics”, in *Proceedings of Workshop on Structured Adaptive Mesh Refinement Grid Methods*, eds. S. Baden, N. Chrisochoides, D. Gannon & M. Norman, IMA Conference Series, Springer Verlag, in press.
- Bryan, G. L. & Norman, M. L. 1997b. “Simulating X-ray Clusters with Adaptive Mesh Refinement”, in *The 12th ‘Kingston Meeting’: Computational Astrophysics*, eds. D. A. Clarke & M. J. West, PASP Conference Series Vol. 123, 363.
- Neeman, H. J. 1996. “Autonomous Hierarchical Adaptive Mesh Refinement for Multiscale Simulations”, Ph. D. thesis, Dept. of Computer Science, University of Illinois at Urbana-Champaign.
- Neeman, H. J. & Norman, M. L. 1997. “HAMR: The Hierarchical Adaptive Mesh Refinement System”, in *Proceedings of Workshop on Structured Adaptive Mesh Refinement Grid Methods*, eds. S. Baden, N. Chrisochoides, D. Gannon & M. Norman, IMA Conference Series, Springer Verlag, in press.

6 Bibliography

- [1] Amza, C. et al. 1996. IEEE Comput. 29/2, 18.
- [2] Berger, M. J. & Colella, P. 1989. J. Comp. Phys., 82, 64.
- [3] Bryan, G. L., Norman, M. L., Stone, J. M., Cen, R. & Ostriker, J. P. 1995. Comp. Phys. Comm., 89, 149.
- [4] Bryan, G. L. & Norman, M. L. 1997a. in *Proceedings of Workshop on Structured Adaptive Mesh Refinement Grid Methods*, eds. S. Baden, N. Chrisochoides, D. Gannon & M. Norman, IMA Conference Series, Springer Verlag, in press.
- [5] Bryan, G. L. & Norman, M. L. 1997b. in *The 12th 'Kingston Meeting': Computational Astrophysics*, eds. D. A. Clarke & M. J. West, PASP Conference Series Vol. 123, 363.
- [6] Colella, P. & Woodward, P. R. 1984. J. Comp. Phys., 54, 174.
- [7] Neeman, H. J. 1996. Ph. D. thesis, Dept. of Computer Science, University of Illinois at Urbana-Champaign.
- [8] Neeman, H. J. & Norman, M. L. 1997. in *Proceedings of Workshop on Structured Adaptive Mesh Refinement Grid Methods*, eds. S. Baden, N. Chrisochoides, D. Gannon & M. Norman, IMA Conference Series, Springer Verlag, in press.
- [9] Villumsen, J. V. 1989. Ap. J. Supp., 71, 407.